NetCache NNTP (NetNews) Caching

Karl L. Swartz | Network Appliance | TR 3059

TECHNICAL REPORT

Table of Contents

[TR3059]

1. Introduction

Network Appliance filers (file server appliances) have been the cornerstone of high-performance netnews servers for many years. NetApp has built upon this experience to create the first high-performance caching appliance for netnews, providing another increment in scalability and cost reduction for Internet Service Providers. Enterprise customers can better provide this often non-critical service at lower cost with NetCache.

This report begins with an overview of traditional netnews server architectures and the issues involved in scaling them to handle hundreds or thousands of concurrent users before proceeding into a discussion of how NetCache may be deployed to produce a less expensive, yet more scalable netnews service. Following that is a section describing how NNTP is implemented within NetCache, focusing on aspects which significantly differ from other netnews implementations. The report closes with some statistics from real deployments.

2. Server Architecture and Scaling

2.1 Traditional NetNews Server Architecture

For most of the 1990s, INN has been the dominant netnews software used at large installations. Its architecture poses substantial challenges for most file systems when handling the *news spool* for non-trivial news feeds. NetApp filers have proven to be equal to this challenge and thus are frequently used as a key component of a high-performance netnews server.

A netnews server's work is split into two major parts. One is maintaining the information, which is dominated by news feeds (outgoing as well as incoming) but also includes expiration of old articles and other housekeeping chores. The other part is serving news to users. While some netnews servers handle only transit traffic, with no users being served directly, all need data and thus must handle a feed.

The key components of INN for these tasks are `innd` (InterNetNews Daemon) and `nnrpd` (NetNews Reader Protocol Daemon, a misnomer since the NNTP protocol encompases commands for both feeds and readers -- there is no separate reader protocol). Scaling to hundreds or thousands of concurrent users reading news (readers) is a challenge because each one requires a separate `nnrpd` process, each of which typically has a working set of about 1 MB. Fortunately, these processes do not need to synchronize with `innd` and thus can be run on any machine which can see the news spool and related data. An NFS-mounted filer enables this scaling by making the data visible to multiple application servers, one of which runs `innd` while the others run the `nnrpd` processes using read-only NFS mounts.

Eventually, even a NetApp filer reaches the limits of its performance. Further scaling can be accomplished by splitting the news data over several filers, resulting in the configuration illustrated in Figure 1. This has its limits, though, since there are only a few reasonable places to split the data. Also, while higher performance demands more disk spindles in addition to other filer resources, the database isn't all that large (about 22.2 GB per day as of March 1999). This approach can thus lead to many large disks each holding relatively little data.

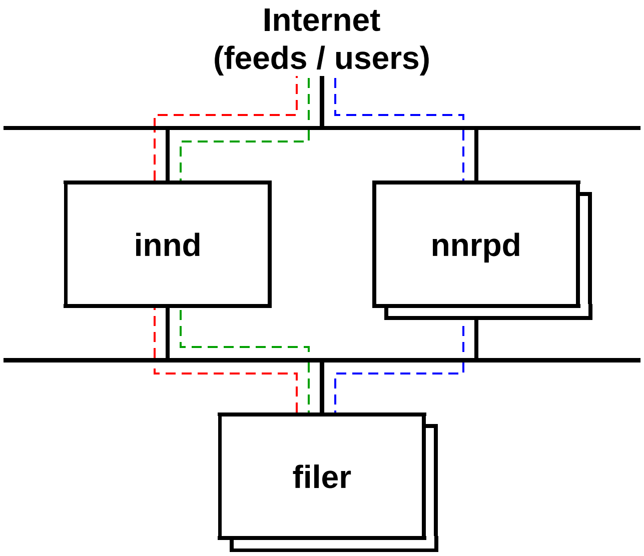

Figure 1: Typical netnews server using INN and filers

2.2 Re-designed News Spool Architectures

The biggest challenge with the traditional INN news spool is that each article is stored in a separate file, resulting in millions of relatively small files. Most file systems don't handle this terribly well, so one approach to the scaling problem has been to change the software to use a few large files, each containing a large number of articles. One implementation of this idea is a set of patches to INN, subsequently incorporated in the INN 2.0 distribution, though this solution still suffers from the `nnrpd` resource issues.

Another popular solution is bCandid's Typhoon software. Typhoon makes much more efficient use of application server resources, and thus can serve many more users than can INN without resorting to multiple application servers. Using a filer further extends the abilities and performance of a single server. Filers can't enable the split of tasks as with INN, however, so scaling requires either a larger and considerably more expensive application server or multiple semi-independent servers with the accompanying management costs.

2.3 Replicated NetNews Servers

Once the limits of a single netnews server have been reached, the next step is to have replicated servers, involving expensive duplication of equipment and administrative effort. One server is configured as a *master* for the purpose of assigning article numbers, with the others being *slaves*, taking their article numbering cues from the Xref: headers generated by the master.

With geographic separation of the servers, this solution also provides for disaster recovery. If the volume of user requests at each location exceeds the volume of a news feed, backbone bandwidth savings can be obtained as well. On the other hand, geographic diversity compounds the administrative costs of replicated servers.

Partial replicas can be implemented to somewhat mitigate the costs of duplicate equipment, primarily storage. There are several ways of accomplishing this: partial feeds to the slave servers, and shorter article retention (expiration) times on the slaves. In either case, maintaining a consistent view of the service to users requires a mechanism, such as Typhoon's server chaining, which permits slaves to know about all articles on the master, and to be able to fetch requested articles from the master if they are not available locally.

2.3.1. Partial feeds

A partial feed may be sent to the slave server, carrying only the newsgroups which are expected to be read by users who use the slave server. This further reduces backbone bandwidth usage -- if the predicted newsgroups cover most requests. Otherwise, if the slave doesn't cache articles from unexpected newsgroups, repeated fetches can potentially *increase* backbone usage. In addition, the selection of newsgroups is usually a manual process, requiring further administrative effort.

2.3.2. Short article retention

Storage costs can be reduced somewhat by retaining articles for a shorter period of time (i.e., expiring them faster) on the slave than on the master. Since most article read requests occur within 3-4 days after the arrival of the article at the server, this can be very effective for sites wishing to offer articles for serveral weeks or more -- the rare requests for older articles can be served from the master.

2.4 NetCache

NetCache offers a better way of scaling netnews servers. NetApp's appliance software design is inherently more efficient [TR-3001], resulting in a netnews server which can handle many more connections per system than conventional netnews software running on a general purpose computer. Fewer systems are therefore required. An additional benefit is faster response to the users -- improved quality-of-service.

2.4.1. Low cost of administration

In addition to greater efficiency, the appliance architecture results in simpler administration. Upgrades are simple and fast, and configuration is easy.

Caching further enhances this effect since there is no need to carefully manage disk space to accomodate short-term bursts and relentless long-term growth in netnews traffic [Swartz1993,Noll1999] -- allowing one to focus these efforts on one central server while the caches handle high user loads with minimal attention. Combined with unit reduction, NetCache dramatically reduces the administrative cost of providing a netnews service.

2.4.2. Reduced bandwidth costs

Another effect of easier administration is that it is more feasible to deploy netnews services in remote locations. The benefit of this to the service provider is that bandwidth costs can be greatly reduced. Users benefit, too, because their requests traverse less of the network and the resulting lower round-trip time improves performance.

3. Deploying NetCache for NetNews

3.1 NetCache as a Server Front-End

One way of deploying NetCache is in the data center as an additional reader server. In effect, it takes over the role of the `nnrpd` servers in an INN-based netnews server as depicted in Figure 2. NetCache still requires a reader server for its own use, but one NetCache appliance as a front-end can replace many `nnrpd` servers while placing only a modest load on the back-end server, allowing substantial unit reduction. In a server based on Typhoon, NetCache reduces the need for expensive server hardware while also enabling the incremental scaling of the netnews service.

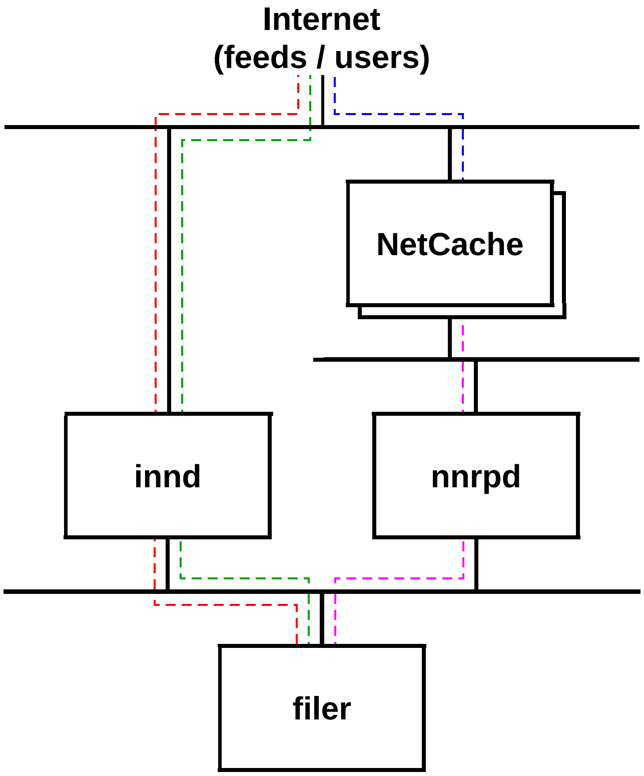

Figure 2: NetCache deployed as part of a central netnews server

3.2 NetCache in remote POPs

The ease of administration of NetCache makes it feasible to deploy caches in unstaffed, remote points-of-presence (POPs). Bandwidth savings and improved service to the user are a direct benefit. The same NetCache appliance can also provide Web caching, adding further bandwidth savings with no additional equipment.

3.3 NetCache in Secondary Data Centers

Smaller caches inevitably result in lower cache hit rates, and thus somewhat higher bandwidth utilization. ISPs covering a large geographic area may find substantial backbone bandwidth savings by interposing large regional caches between the smaller caches in POPs and the central data center. For example, an ISP based in New York with a substantial customer base on the West Coast might benefit from placing a secondary server in San Francisco.

This can be done using slave servers, but the ease of administration of NetCache makes it easier to deploy, just as with POPs. Omitting unwanted articles (such as spam which is quickly cancelled, before ever being requested by users) results in added bandwidth savings relative to a slave server.

4. NetCache NNTP Implementation Details

4.1 NNTP Command Subset for a Caching Server

The NNTP protocol was first formally documented in RFC 977. Many extensions were subsequently implemented, some of which have become de facto standards while others have remained rare.

NNTP is comprised of two nearly non-overlapping subsets. The larger portion of the commands are used by client reader software .(browsers) to access netnews on a server. A smaller subset is used to facilitate exchange of news feeds between cooperating servers. Since NetCache does not handle feeds, it only implements the reader-oriented subset of NNTP. Most of these commands are directly implemented, either using data cached from the server or, in a few special cases, simply processing the command locally. Table 1 lists the RFC 977 commands implemented within NetCache.

Command	Action
`ARTICLE`	served from cached data
`BODY`	served from cached data
`GROUP`	served from cached data
`HEAD`	served from cached data
`HELP`	served locally
`LAST`	served from cached data
`LIST`	served from cached data
`NEXT`	served from cached data
`QUIT`	served locally
`STAT`	served from cached data

Table 1: RFC 977 commands implemented by NetCache

Some RFC 977 commands require a complete article history database (e.g., `NEWNEWS`) or otherwise are not possible for a cache to directly implement (e.g., `POST`). These commands, which are simply proxied to the server, along with the feed-specific commands, are enumerated in Table 2.

Command	Action

IHAVE	rejected (only for feeds)
NEWGROUPS	proxied to server
NEWNEWS	proxied to server
POST	proxied to server

Table 2: Proxied or inapplicable RFC 977 commands

Of the post-RFC 977 extensions to NNTP, the XOVER command, used to access the News OverViews (NOV) database, is the most widely used and has become crucial to the efficient functioning of modern netnews clients and servers -- and of NetCache itself (see §4.3). It is implemented using cached information.

AUTHINFO is also important for many ISPs. NetCache currently implements the original form of this command if a RADIUS server is configured. (Other forms of AUTHINFO may be implemented in the future, depending on customer requirements.)

Other NNTP extensions are comparively rare in practice, but are crucial for those clients which do use them. NetCache may serve them from cached data in the future, but at present they are simply proxied.

Table 3 details the extensions implemented by NetCache and how they are implemented.

Command	**Action**
AUTHINFO USER AUTHINFO PASS	via RADIUS if configured
LIST *args* ...	proxied to server
LISTGROUP	served from cached data
MODE READER	served locally
XGTITLE	proxied to server
XHDR	proxied to server
XOVER	served from cached data
XPAT	proxied to server

Table 3: Common extensions recognized by NetCache

4.2 Articles Stored by Message-id

Individual articles are stored via the same NetCache storage manager developed and proven for Web caching, with the object key being derived from the article's message-id instead of the URL used for Web pages. This differs from traditional netnews servers, which store articles using a filename based on the newsgroup name and article number within the newsgroupd, requiring multiple primary keys via which the article may be accessed, usually implemented with hard links.

4.3 News OverView (NOV) is Keystone

The vast majority of article requests from clients are made using the article number within the current newsgroup, not the message-id. NetCache uses an efficient local implementation of the News OverView (NOV) database to provide the mapping from (newsgroup, article-number) pair to the message-id-based cache key.

5. Case Studies

Detailed case studies of typical customer sites will be added to this document in a future revision.

5.1 Typical U.S. Dialup ISP

The baseline case for discussing netnews service is an ISP providing dialup (modem) service within the U.S., as well as Canada and other areas in which residential telephone service is offered with unmeasured local calling, i.e., a local call lasting for hours costs no more than a one-minute local call.

Netnews servers at this type of site handle many concurrent connections, each one doing relatively little work. This places substantial demands on the network stack. Memory can also be an issue unless the server is carefully written to keep the memory footprint for each client as small as possible. NetCache has been designed with these requirements firmly in mind.

5.2 High-speed ISP

ISPs providing high-speed connectivity via services such as DSL or cable modems place a somewhat different load on their netnews servers. Higher speeds mean requests complete more quickly, so one client can generate more requests per unit of time than can dialup clients. Memory conservation is less important than efficiently handling many requests at relatively high bandwidths.

5.3 European ISP

In Europe and other parts of the world, all residential phone service is measured, so dial-up users want to download netnews as quickly as possible and read it off-line. Beyond keeping their connections fully utilized, unlike U.S. and similar readers who often create pauses while they read an article, the software typically used for off-line reading uses the `NEWNEWS` command to get a list of articles which are then requested by message-id. On-line readers typically request articles by article number after selecting a newsgroup. Some netnews software performs quite poorly when articles are requeted by message-id, but since NetCache used the message-id as its object key it handles both off-line and on-line readers with comparable efficiency.

5.4 Enterprise (or Small ISP) Outsourced NetNews

The fixed cost of running a news feed (WAN bandwidth, storage, and administration) is quite high. If there are not a large number of users over which to amortize this cost, outsourcing the service can be very attractive. One pitfall of this approach is that the server may be distant in terms of network topology, resulting in poor response time for users. Another problem is that users who look at the same articles will cause those articles to be transferred over the WAN multiple times, potentially resulting in higher WAN bandwidth than with a carefully tuned local server.

NetCache addresses both of these problems, while still allowing the high-cost job of running a news server to be outsources.

6. Conclusion

The NetCache implementation of NNTP offers a way to build scalable, high-performance netnews servers with dramatically reduced equipment and administrative cost compared to traditional netnews servers. The ease of administration additionally enables deployment of caches in remote, unstaffed locations, reducing bandwidth costs while providing faster service to the user community.

7. References

[Ahola1996]
Esa Ahola, "Re: What are other people doing for I/O subsystems?," *news.software.nntp Usenet newsgroup*, 23 October 1996.

[Ahola1997]
Esa Ahola, "Re: Netapp for Usenet news," *toasters mailing list*, 5 March 1997.

[Barber1998]
Stan Barber, *Common NNTP Extensions*, August 1998.

[Barber1999]
Stan Barber, *Network News Transfer Protocol*, August 1999.

[Chank1996]
Anawat Chankhunthod, Peter Danzig, et. al., "A Hierarchical Internet Object Cache," *Proceedings of the 1996 USENIX Annual Technical Conference*, pp. 153-163, San Diego, January 1996.

[Christenson1997]
Nick Christenson, David Beckemeyer, and Trent Baker, "A Scalable News Architecture on a Single Spool," *;login:*, June 1997.

[Fritchie1997]
Scott Lystig Fritchie, "The Cyclic News Filesystem: Getting INN To Do More With Less," *Proceedings of the 11th USENIX Large Installation System Administration Conference (LISA XI)*, pp. 99-111, San Diego, October 1997.

[INN2.2]
inn.conf(5) man page, *INN 2.2*.

[Noll1999]
Landon Curt Noll, "All the News that's fit to NNTP", *Stanford University CS 545*, Palo Alto, California, 4 June 1999.

[RFC-977]
Brian Kantor, Phil Lapsley, *RFC 977: Network News Transfer Protocol*, February 1986.

[Salz1992]
Rich Salz, "InterNetNews: Usenet transport for Internet sites," *Proceedings of the 1992 Summer USENIX Technical Conference*, pp. 93-98, San Antonio, Texas, June 1992.

[Swartz1993]
Karl L. Swartz, "Forecasting Disk Resource Requirements for a Usenet Server," *Proceedings of the 7th USENIX Large Installation System Administration Conference (LISA VII)*, pp. 101-108, Monterey, California, November 1993.

[Swartz1996]
Karl L. Swartz, "The Brave Little Toaster Meets Usenet," *Proceedings of the 10th USENIX Large Installation System Administration Conference (LISA X)*, pp. 161-170, Chicago, October 1996.

[TR-3001]

Dave Hitz, *An NFS File Server Appliance (TR-3001)*, Network Appliance, Santa Clara, California, March 1995.

[TR-3029]

Peter Danzig, *NetCache Architecture and Deployment (TR-3029)*, Network Appliance, Santa Clara, California, 1998.

[Typhoon1999]

Typhoon NewsServer, bCandid, Boulder, Colorado, 1999.

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089
www.netapp.com